## REMARKS

Claims 1-12 and 14-23 are pending in this application. In the outstanding Office Action, claims 1-12 and 14-23 stand rejected under 35 U.S.C. § 103(a) as being allegedly unpatentable over United States Patent Application Publication No. 2003/0101169 ("Bhatt") in view of United States Patent No. 6,836,778 ("Manikutty"). Applicants respectfully traverse.

### *Interview Summary*

Applicants' undersigned representative, Mr. Eiferman, and Examiner Susan Rayyan participated in a telephonic interview on December 13, 2007, during which the above claim amendments were discussed. Examiner Rayyan agreed to reevaluate the rejections in view of the above claim amendments.

### *Rejections under 35 U.S.C. § 103*

Claims 1-12 and 14-23 stand rejected under 35 U.S.C. § 103(a) as being allegedly unpatentable over United States Patent Application Publication No. 2003/0101169 ("Bhatt") in view of United States Patent No. 6,836,778 ("Manikutty"). Applicants respectfully traverse.

### Claims 1-12, 14 and 15

Independent claims 1 and 10 are directed to techniques for processing a query including an extensible markup language (XML) based expression with instructions to modify data that is stored in a node of an XML schema in a database. As part of these techniques, an abstract syntax tree corresponding to the expression is transformed into a unified tree including XML based algebra operations. The XML based algebra operations in the unified tree are mapped to enhanced relational algebra based extensible markup language modification operations in a relational tree. The query is then executed by modifying data that is stored in the node of the extensible markup language schema in the database in accordance with the relational tree.

Bhatt discloses an (1) XML store engine for storing XML data; and (2) an XQL engine for querying XML data that is stored in the XML store engine. However, the Bhatt XQL engine executes only certain types of queries. In particular, the Bhatt XQL engine *only* executes queries that *read* data from the XML store engine. The Bhatt XQL engine does *not* execute queries that *modify* data in the XML store engine. Put more simply, Bhatt does not teach or suggest that the XQL engine (*i.e.,* the query processor) can *modify* data in the XML store engine. Thus, Bhatt does not teach or suggest "enhanced relational algebra based XML modification operations" as recited in independent claims 1 and 10.

Manikutty discloses modifying XML data. However, Manikutty does not teach or suggest modifying XML data using enhanced relational algebra operations. Rather, Manikutty discloses that an in memory representation if the XML data is generated using a data object model (DOM) data structure or a schema based data structure (Col. 12, ll. 9-25). In fact, there is no mention of relational algebra anywhere in Manikutty. Thus Manikutty also fails to teach or suggest "*enhanced relational algebra based* XML modification operations" as recited in independent claims 1 and 10.

Thus, the cited references do not teach or suggest "enhanced relational algebra based XML modification operations," as recited in independent claims 1 and 10. Accordingly, Applicants respectfully submit that independent claims 1 and 10 are patentable over the cited references. Applicants further submit that claims 2-9, 11, 12, 14 and 15 are patentable at least by reason of their dependency.

## Claims 16-23

Independent claims 16 and 20 recite that a query is parsed to yield an XML based expression. A query plan is then generated for the XML expression including enhanced relational algebra expressions with a nested table abstraction operation. The query is then executed based on the query plan by performing an action on an extensible markup language schema by using the nested table abstraction operation to establish a parent to descendent relationship among instances of nodes in the extensible markup language schema without compiling separate lists.

Bhatt discloses an XQL engine that can execute queries that request data from parent and descendant nodes. However, in Bhatt, the such queries are executed by compiling a list

of instances of the parent node, a list of instances of the descendant node, and then comparing the two separate lists to establish the corresponding descendant instances for each parent instance (Bhatt, ¶0129-0145). Thus, Bhatt does not teach or suggest a nested table abstraction operation to establish a parent to descendent relationship among instances of nodes in the extensible markup language schema without compiling separate lists. Manikutty similarly fails to teach or suggest this feature.

Thus, the cited references do not teach or suggest "executing the query based on the query plan by using the nested table abstraction operation to establish a parent to descendent relationship among instances of nodes in the extensible markup language schema without compiling separate lists," as recited in independent claim 16 (or similar features from independent claim 20). Accordingly, Applicants respectfully submit that independent claims 16 and 20 are patentable over the cited references. Applicants further submit that claims 17-19 and 21-23 are patentable at least by reason of their dependency. Thus, reconsideration and withdrawal of the 35 U.S.C. § 103(a) rejections are respectfully requested.

<u>**CONCLUSION**</u>

  In view of the above remarks, Applicants respectfully submit that the present application is in condition for allowance. Applicants respectfully request reconsideration of the present application.

Date: January 7, 2008         /Kenneth R. Eiferman/
                     Kenneth R. Eiferman
                     Registration No. 51,647

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439